

Attorney Docket No. RSW920030177US1  
Serial No. 10/705,555  
Response to Office Action mailed February 20, 2007

### III. AMENDMENTS TO THE SPECIFICATION

Please amend paragraph [1] as follows:

- [1] This application incorporates by reference the following commonly-assigned and co-pending U.S. Patent Applications, filed on November 10, 2003: United States Application Serial Number 10/704,836, entitled AUTOMATIC PARALLEL NON-DEPENDENT COMPONENT DEPLOYMENT; and United States Application Serial Number 10/705,525, entitled ~~PRE-DEPLOYMENT COMPONENT HOSTING ENVIRONMENT ANALYZER~~ INSTALLATION AND REMOVAL OF SOFTWARE COMPONENTS ACROSS ENTERPRISE RESOURCES.

Attorney Docket No. RSW920030177US1  
Serial No. 10/705,555  
Response to Office Action mailed February 20, 2007

Please amend paragraph [2] as follows

[2] An additional issue is raised due to the almost infinite number of combinations of software settings and configurations on multiple hosts with multiple parameters. Such complexity makes it extremely difficult for an administrator to devise reliable test plans to insure the validity of change to software within an enterprise. Thus, seemingly harmless upgrades, patches or new software may wreak havoc on an enterprise infrastructure. Existing software may unintentionally be compromised or corrupted by additional software or software updates. It will be appreciated that such unforeseen consequences may cause part or even all of a business's enterprise system to fail. For example, a new Java JAVA Software Development Kit (SDK) is deployed each time an application, which uses ~~Java~~JAVA, is deployed. Although the ~~Java~~-JAVA SDKs are supposed to be back-compatible they are not. Furthermore, developers commonly use both ~~Sun~~SUN and IBM ~~Java~~JAVA SDKs, introducing a number of incompatibilities. That is, ~~Java~~-JAVA applications which were functional under SUN ~~Java~~JAVA version 1.3.1, for example, might not work properly under SUN ~~Java~~JAVA 1.4.1 or IBM ~~Java~~JAVA 1.3.1.

Attorney Docket No. RSW920030177US1  
Serial No. 10/705,555  
Response to Office Action mailed February 20, 2007

Please amend paragraph [3] as follows:

[3] The ~~Java~~JAVA SDK incompatibilities described above present one of the more common problems in ~~Java~~JAVA 2 Platform Enterprise Edition (J2EE) environments. However, although very harmful, this is a relatively simple problem to detect. More complicated problems are presented at the operating system (OS) and compiler levels. Frequently at the OS level there may be incompatibilities between different versions of an OS kernel and certain applications. For instance, IBM ~~Java~~JAVA SDK version 1.4.1 runs only with a ~~Linux~~LINUX kernel 2.2.5 or less, while the current ~~Linux~~LINUX kernel on ~~Redhat~~REDHAT ~~Linux~~LINUX is 2.5. Thus a new deployment will likely update the kernel and consequently perturb the functionality of the ~~Java~~JAVA Virtual Machine (JVM) and consequently all applications that use the JVM. A similar problem might occur with OS patches.

Attorney Docket No. RSW920030177US1  
Serial No. 10/705,555  
Response to Office Action mailed February 20, 2007

Please amend paragraph [28] as follows:

[28] The present invention employs a "semantic model" described more fully in commonly assigned and co-pending U.S. Patent Serial Number ~~\_\_\_\_\_10/727,011~~, filed ~~\_\_\_\_\_12/3/2003~~, ~~IBM Disclosure RSW8-2003-0414~~, entitled eREGISTRY RECORDER AND ROLL BACK, hereby incorporated by reference. Such a model, generated by the developer and included in the installation package, provides a "taxonomy" of all software components of interest, such as all software which IBM, for example, produces or uses. The model comprises a set of entries for each application, component and sub-component being installed (hereinafter collectively referred to as "components"). The model includes:

- references or links among components indicating their deployment dependencies;

- entries indicating what other components are necessary for the proper operation of each component being installed; and

- entries indicating incompatibilities with other components likely to have been previously installed.

Attorney Docket No. RSW920030177US1  
Serial No. 10/705,555  
Response to Office Action mailed February 20, 2007

Please amend paragraph [30] as follows:

[30] The semantic model is a data structure stored in a knowledge base (as more fully described in commonly-assigned and co-pending U.S. Patent Serial Number 10/726,192, filed 12/2/2003, ~~IBM Disclosure Number RSW8-2003-0443~~, entitled HOSTING ENVIRONMENT ABSTRACTION AGENTS, hereby incorporated by reference). The data structure need not be any particular structure; examples of possible structures include (but are not limited to) a flat file, a database, an object model, etc. The component semantic model is generated by the developer and may be bundled with the deployment package or accessed from a remote site during installation. In the event that deployment is to occur across domains, the model may be augmented with a list of target machines on which components will be installed.

Attorney Docket No. RSW920030177US1  
Serial No. 10/705,555  
Response to Office Action mailed February 20, 2007

Please amend paragraph [35] as follows:

[35] The present invention also identifies potential component conflicts by implementing a pre-deployment hosting environment analyzer. Again the semantic model for software components is employed which captures the topology of software components at different levels of detail as well as capturing complex relationships among components. The deployed components on the target are recorded in the eRegistry. The installation is as follows: as soon as an eReadme file is available to deploy (an eReadme captures the information about the components that are to be deployed), the eRegistry is examined and the knowledge base (as more fully described in commonly-assigned and co-pending U.S. Patent Serial No. 10/725,612, filed 12/2/2003, entitled OPTIMAL COMPONENT INSTALLATION) is accessed to download metadata about the relationship among the components to be installed and the components existing in the target. Next, the relationship data is analyzed so appropriate action may be taken in the event that a conflict is identified. For example, the installation may continue or the user may be alerted of the possible conflict. In the event installation continues, an entry may be recorded in a log for later reference. As soon the software is deployed on the target, the target eRegistry is updated with appropriate installation information.